

WE CLAIM:

Sub Q1

1. A Turing complete computer implemented learning method comprising the steps of:

(a) providing an indirectly executable computer program including:

a first instruction that points to and designates machine code stored in a memory as data;

a second instruction that points to and designates machine code stored in the memory as at least one directly executable function;

a third instruction that alters machine code pointed to by the first instruction; and

a fourth instruction that executes machine code pointed to by the second instruction; and

(b) controlling the program to perform the steps of:

(b1) creating and storing a machine code entity including at least one directly executable function having a directly executable branch instruction in a memory;

(b2) executing the second instruction to point to the entity;

(b3) executing the third instruction using training data to produce a result;

(b4) processing the result using a learning algorithm;

(b5) executing the first instruction to point to the entity;

(b6) altering the entity using an alteration algorithm to include at least one other directly executable function; and

(b7) repeating steps (b3) to (b6) until an end criterion is reached.

2. A method as in claim 1, in which the branch instruction is a recursive instruction.

3. A method as in claim 1, in which the branch instruction is a leaf function call.

4. A method as in claim 1, in which the branch instruction is an external function call.

5. A method as in claim 1, in which the branch instruction is a subroutine call.

6. A method as in claim 1, in which:
 the first instruction points to and designates machine code stored in a memory as data by casting a pointer to the entity as a data pointer; and
 the second instruction points to and designates machine code stored in the memory as a directly executable function by casting said pointer as a function pointer.

7. A method as in claim 1, in which:
 said at least one directly executable function as comprises a return instruction; and
 step (b6) comprises preventing the return instruction from being altered.

Rule 1,126

8. A method as in claim 1, in which:
 step (b1) comprises creating and storing the entity as an array of directly executable functions;
 step (b2) comprises executing the second instruction to point to the array; and
 step (b5) comprises executing the first instruction to point to the array.

9. A method as in claim 9, in which:
 each function includes at least one directly executable instruction; and
 the functions can include different numbers of directly executable instructions.

~~10~~
~~11.~~ A method as in claim 10, in which the functions have a maximum length.

~~11~~
~~12.~~ A method as in claim 9, in which at least one of the functions includes a subroutine.

~~12~~
~~13.~~ A method as in claim 1, in which said at least one function comprises a subroutine that is called by the branch instruction.

~~13~~
~~14.~~ A method as in claim 1, in which said at least one function comprises a plurality of subroutines that can call each other.

~~14~~
~~15.~~ A method as in claim 1, in which:
said at least one function comprises a main function and a plurality of subroutines which have headers respectively; and

step (b1) comprises initializing the headers such that some of the subroutines cannot call others of the subroutines.

~~15~~
~~16.~~ A method as in claim 1, in which:
said at least one function comprises a main function and a subroutine having headers respectively; and
step (b1) comprises initializing the header of the subroutine such that it can call itself and thereby perform recursion.

~~16~~
~~17.~~ A Turing complete computer learning system comprising:
memory means for storing an indirectly executable computer program including:

a first instruction that points to and designates machine code stored in a memory as data;

a second instruction that points to and designates machine code stored in the memory as at least one directly executable function;

a third instruction that alters machine code pointed to by the first instruction; and

a fourth instruction that executes machine code pointed to by the second instruction; and

processing means for executing the program;

the processing means, memory means and program operating in combination for performing the steps of:

(a) creating and storing a machine code entity including at least one directly executable function having a directly executable branch instruction in the memory means;

(b) executing the second instruction to point to the entity;

(c) executing the third instruction using training data to produce a result;

(d) processing the result using a learning algorithm;

(e) executing the first instruction to point to the entity;

(f) altering the entity using an alteration algorithm to include at least one other directly executable function; and

(g) repeating steps (b) to (f) until an end criterion is reached.

48 ~~18~~ 18. A system as in claim 17, in which the branch instruction is a recursive instruction.

49 ~~19~~ 19. A system as in claim 17, in which the branch instruction is a leaf function call.

50 ~~20~~ 20. A system as in claim 17, in which the branch instruction is an external function call.

51 ~~21~~ 21. A system as in claim 17, in which the branch instruction is a subroutine call.

21
~~22~~. A system as in claim 17, in which:

the first instruction points to and designates machine code stored in a memory as data by casting a pointer to the entity as a data pointer; and

the second instruction points to and designates machine code stored in the memory as a directly executable function by casting said pointer as a function pointer.

22

~~23~~. A system as in claim 17, in which:

said at least one directly executable function as comprises a return instruction; and

step (b6) comprises preventing the return instruction from being altered.

23

~~24~~. A system as in claim 17, in which:

step (b1) comprises creating and storing the entity as an array of directly executable functions;

step (b2) comprises executing the second instruction to point to the array; and

step (b5) comprises executing the first instruction to point to the array.

24

~~25~~. A system as in claim 24, in which:

each function includes at least one directly executable instruction; and

the functions can include different numbers of directly executable instructions.

25

~~26~~. A system as in claim 25, in which the functions have a maximum length.

26

~~27~~. A system as in claim 24, in which at least one of the functions includes a subroutine.

27

~~28~~. A system as in claim 17, in which said at least one function comprises a subroutine that is called by the branch

instruction.

~~28~~
~~29.~~ A system as in claim 17, in which said at least one function comprises a plurality of subroutines that can call each other.

~~29~~
~~30.~~ A system as in claim 17, in which:
said at least one function comprises a main function and a plurality of subroutines which have headers respectively; and

step (b1) comprises initializing the headers such that some of the subroutines cannot call others of the subroutines.

~~30~~
~~31.~~ A system as in claim 17, in which:
said at least one function comprises a main function and a subroutine having headers respectively; and
step (b1) comprises initializing the header of the subroutine such that it can call itself and thereby perform recursion.

~~31~~
~~32.~~ A system as in claim 17, in which:
the memory means comprises a main memory and a processor memory that is part of the processor; and
step (a) comprises storing the entity in the processor memory

~~32~~
~~33.~~ A system as in claim 32, comprising an integrated circuit chip, in which the processor and the processor memory are formed on the chip.

~~33~~
~~34.~~ A system as in claim 33, in which at least a portion of the program is stored processor memory.

~~34~~
~~35.~~ A method as in claim 34, in which:
the processor memory comprises a non-volatile memory section; and

said at least a portion of the program is stored in the non-volatile memory section.

add a³7